Improvement of UVM IP Validation using Portable Stimulus (PSS)

Robert R Martin, Intel Corporation
Alan M Curtis, Intel Corporation
Gopinath Narasimhan, Synopsys
Qingwei Zhou, Intel Corporation

accellera SYSTEMS INITIATIVE

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

INTRODUCTION

Portable Stimulus **(PSS)** is typically pitched as the solution to port test intent across different validation platforms. This paper details how one could adopt PSS into their existing UVM IP validation regime for efficient Scenario Generation, Control and Reproducibility. Test stimulus can be distributed across the PSS tool and the UVM testbench framework. Test scenario creation and orchestration can either (1) entirely be done in PSS, where handoff happens at the UVM atomic sequences OR (2) in a hybrid PSS-UVM setup, where handoff is at the UVM virtual sequence. This PSS to UVM handoff point could be a sliding scale, tailored to any IP or SoC validation goals. Stimulus intent could be modeled in DSL**,** and a PSS processing tool can generate stimulus/tests from the model. The coverage from PSS tests can be analyzed prior, and the best tests selected for UVM mapped simulations.

An Intel Memory-IP library testplan was executed and other validation capabilities, tested using a PSS/DSL's OOP & AOP constructs. This paper highlights this unique PSS flow, leveraging Synopsys' *VC Portable Stimulus **(VCPS)*** tool capabilities to supplement a robust framework, facilitating deterministic & reproducible randomness in continuous integration systems (Gatekeeper) – saving Intel ***dollars***!

PROBLEM STATEMENT / OBJECTIVES

Drawbacks of conventional UVM based Functional Verification:

Test Redundancy: SV random seed-based stimulus may partly or wholly retake the same values across different seeds. This results in redundant simulations that wastes compute cycles with no coverage improvement.

Test Reproducibility: SV UVM test-suites and their associated testbench counterparts may change multiple times throughout a project lifecycle. As code changes across validation milestones and bug fixes, test behavior may change. The same seed may produce different stimulus across different versions of the simulation, making reproducibility difficult and loss of feature-specific tests.

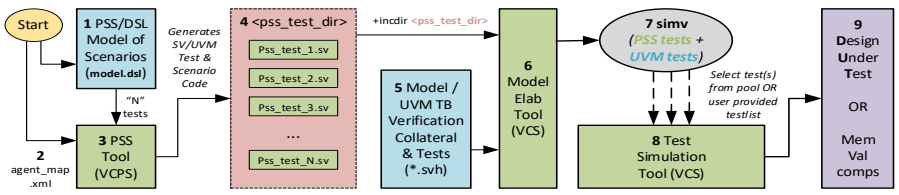
UVM sequence transition & branching coverage: Modeling and tracking coverage to gain visibility at a sequence transition level in an easy, scalable and effective way is non-existent. Focused UVM tests are only as good as the test-writer's ability or how a scenario is depicted in the test plan. There is no real indicator to a validator that all paths or trajectories a test *could take* have been taken, tested and most importantly covered.

This paper showcases how PSS can be used in an existing UVM framework to overcome the above issues by generating unique and reproducible tests from validation scenarios. Pre-elab / pre-sim coverage, can be used as a metric to QA stimulus even before a simulation is run.

RESULTS - I

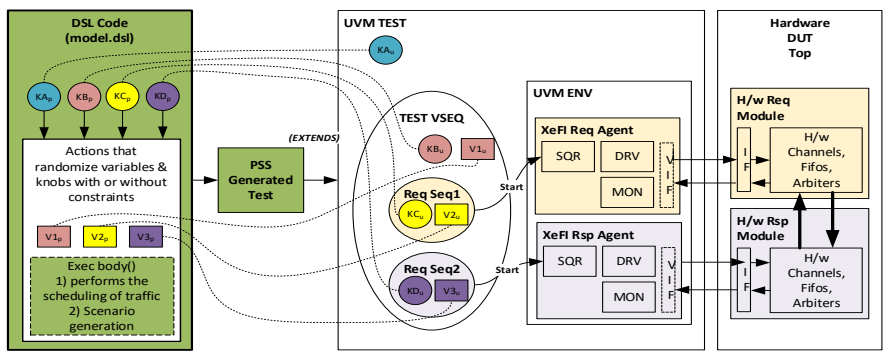
Mode 1: Scenario Orchestration in PSS and UVM handoff at atomic sequence:
In Fig below, DSL Modeled Scenarios (1&2) fed to the VCPS PSS Tool (3), generates multiple tests (4) with unique test paths & random sequences. These are included in the existing UVM Model (5) along with hand-written UVM tests. All tests/code are elaborated using VCS (6). The resulting simv (7) contains all test possibilities. Validator picks test (8) to inject into DUT/IP (9).

These can be fixed-behavior UVM tests or a coverage-report-based PSS test that accurately hits a test plan scenario. Each PSS test, contains fixed intent (modeled in DSL) while UVM only implements the target IP bus protocol. Almost all of randomization happens in PSS, tests are unique by generation, fully reproducible once mapped and provide sequence transition path coverage in addition to functional coverage of stimulus.



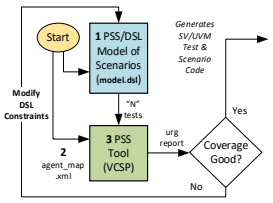
RESULTS - II

Mode 2: PSS-UVM Hybrid Solution, UVM handoff at Virtual Sequence:
The Fig below shows how PSS (K^*_p) knobs control their UVM counterparts (K^*_u). Test flows are generated by VCPS (deterministically) while UVM provides transaction randomization. (***Hybrid solution!!)*** Test writers may thus use PSS without knowledge of the UVM TB. Tests are fully reproducible up to the PSS/UVM control knob level and simulations perform additional controlled randomizations.



RESULTS - III

VCPS URG reports are generated for solve or pre-run-time analysis. PSS scenario model could be iteratively modified, or tool could generate more tests from same scenario to improve stimulus and path coverage. Tool generation times are also shown. Single test template with 1 random cycle takes 0.3s to generate stimulus, while 100 test templates with 100 random variations per template takes only 25.6s. Thus, stimulus coverage indicators can be analyzed prior to launching even a single simulation.



Num Tests	Num Cycles / Test	Total Cycles	Comp Time (s)	Elab Tim (s)	Link Time (s)	Solve Time (s)	Total Gen. Time (s)
1	1	**1**	.616 s	.172 s	.612 s	**0.390 s**	**1.79**
10	1	10	.620 s	.174 s	.603 s	**0.800 s**	2.20
10	10	100	.614 s	.180 s	.602 s	**1.360 s**	2.76
100	10	1,000	.619 s	.177 s	.609 s	**6.690 s**	8.10
100	100	**10,000**	.618 s	.179 s	.605 s	**25.61 s**	**27.00**

Iteration #1: PSS generation coverage for **10 tests**

Summary for Group pss_top::base_fmt_a::fi_pkt_cg

CATEGORY	EXPECTED	UNCOVERED	COVERED	PERCENT
Variables	42	2	40	94.44
Crosses	480	396	84	17.50

Crosses for Group pss_top::base_fmt_a::fi_pkt_cg

CROSS	EXPECTED	UNCOVERED	COVERED	PERCENT	GOAL	WEIGHT	AT LEAST	PRINT MISSING	COMMENT
CR_cmds1	480	396	84	17.50	100	1	1	0	

Iteration #2: PSS generation coverage for **30 tests** ***(Improved)***

Summary for Group pss_top::base_fmt_a::fi_pkt_cg

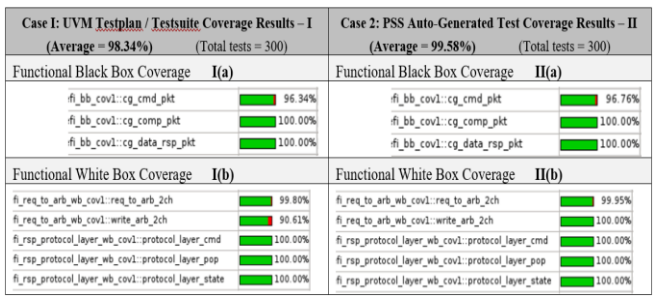
CATEGORY	EXPECTED	UNCOVERED	COVERED	PERCENT
Variables	42	1	41	97.22
Crosses	480	293	187	38.96

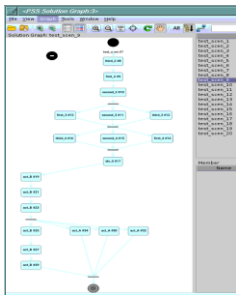
Crosses for Group pss_top::base_fmt_a::fi_pkt_cg

CROSS	EXPECTED	UNCOVERED	COVERED	PERCENT	GOAL	WEIGHT	AT LEAST	PRINT MISSING	COMMENT
CR_cmds1	480	293	187	38.96	100	1	1	0	

CONCLUSIONS

Employing PSS to supplement a UVM framework overcame the shortcomings of a typical SV env setup. 300 automatically generated PSS tests beat the black/white box functional UVM coverage compared to Manual UVM tests, cutting down development time significantly. Tool also provides a visual depiction of the test trajectories previously not seen in simulation.





REFERENCES

PSS - https://www.accellera.org/images/downloads/standards/Portable_Test_Stimulus_Standard_v20.pdf
DSL - https://en.wikipedia.org/wiki/Domain-specific_language
VCPS - https://spdocs.synopsys.com/dow_retrieve/latest/vg/vc_ps/PDFs/vcps_user_guide.pdf
https://www.synopsys.com/content/dam/synopsys/verification/datasheets/costart-portable-stimulus-ds.pdf

Intel Corporation
1900 Prairie City Road,
Folsom, CA,
USA, 95630

Robert.r.martin@intel.com
Alan.m.Curtis@intel.com
Qingwei.zhou@intel.com

Gopinath.Narasimhan@synopsys.com

Synopsys
690 E Middlefield Rd,
Mountain View, CA,
USA, 94043